Model PCL-813

32-Channel Single-Ended
Isolated Analog Input Card

Copyright Notice

This documentation and the software included with this product are copyrighted 1994 by Advantech Co., Ltd. All rights are reserved. Advantech Co., Ltd. reserves the right to make improvements to the products described in this manual at any time without notice.

No part of this manual or software may be reproduced, copied, translated or transmitted, in any form or by any means without the prior written permission of Advantech Co., Ltd. Information provided in this manual is intended to be accurate and reliable. However, Advantech Co., Ltd. assumes no responsibility for its use, nor for any infringements of rights of third parties which may result from its use.

Acknowledgments

PC-LabCard is a trademark of Advantech Co., Ltd. IBM, PC and PC/XT/AT are trademarks of International Business Machines Corporation. MS-DOS, MASM, QuickBASIC, Microsoft C and MS-PASCAL are trademarks of Microsoft Corporation. Intel is a trademark of Intel Corporation. Turbo C and Turbo PASCAL are trademarks of Borland International. MOXA is a trademark of 404 Technologies Inc.

Part No. 2003813030, 4th Edition
Printed in Taiwan, August, 1994

Contents

CHAPTER 1 INTRODUCTION 1

Features 2
Specifications 3
General Specifications 3

CHAPTER 2 INSTALLATION 5

Initial Inspection 6
Switch and Jumper Settings 7
Connector Pin Assignment 9
Plugging the PCL-813 into your PC 10

CHAPTER 3 SIGNAL CONNECTION 11

Analog Input Connection 12

CHAPTER 4 REGISTER STRUCTURE AND FORMAT 13

I/O Port Address Map 14
A/D Data Registers 16
Gain Control Register 17
Multiplexer Scan Register 18
How to Initiate an A/D Conversion 19

CHAPTER 5 SOFTWARE 21
Introduction 22
Parameter Table 22
Parameter Descriptions 27
Function List 28
Function Description 29
Language Interface 31
BASICA 31
GWBASIC (version 3.20) 31
QuickBASIC 4.0 and 4.5 32
Microsoft C 32
Turbo C 33
Borland C++ 33
Microsoft PASCAL 34
Turbo PASCAL 34

APPENDIX A CALIBRATION 35
VR Assignments 36
VR location 37
A/D Calibration 38

APPENDIX B PCLD-881 INDUSTRIAL TERMINATION BOARD 39
Introduction 40
Features 40
Applications 41
Technical Diagrams 42

CHAPTER 1

INTRODUCTION

INTRODUCTION

The PCL-813 32 Channel Single-Ended Isolated Analog Input Card is an easy to use and cost effective IBM PC/XT/AT compatible data-acquisition card. The specifications and the user-friendly software driver of this card make it a popular solution for a wide range of industrial and laboratory applications. Such applications include: measurement of transducer and sensor data, waveform acquisition and measurement, process monitoring, and vibration and transient analysis.

Features

- 32 Single-ended, isolated analog inputs
- 12-Bit resolution A/D conversion
- Programmable analog input ranges: ±5 V, ±2.5 V, ±1.25 V, ±0.625 V, 0~10 V, 0~5 V, 0~2.5 V, 0~1.25 V
- Supports software trigger
- Versatile language drivers including BASIC, PASCAL, C and C++

Specifications

Analog Input (A/D Converter)
Channels: 32 Single-ended with isolation

Resolution: 12-bit, successive approximation

Input Range:
Bipolar: ±5 V, ±2.5 V, ±1.25 V, ±0.625 V, software programmable

Unipolar: 0 ~ 10 V, 0 ~ 5 V, 0 ~ 2.5 V, 0 ~ 1.25 V, software programmable

(Bipolar or Unipolar is selected by JP100)

Converter: AD574 or equivalent, with 25 ms conversion time

Data transfer rate: 25 Kbps maximum, software control only

Isolation voltage: >500 V_{DC} from input to output

Accuracy: 0.01% of reading ±1 LSB

Nonlinearity: ±1 bit maximum

Amplification: x1, x2, x4, x8, software programmable

Trigger mode: By software trigger

Temp. coefficient: ±25 ppm/°C

Overvoltage: Continuous ±30 V maximum

Input impedance: >10 MW

General Specifications

Power consumption: +5 V: 660 mA, typical
+12 V: 140 mA, typical

I/O connector: 37-pin D-type connector (analog input port)

Operating temp.: 0 to 50°C (32 to 122°F)

Storage temp.: -20 to 50°C (-4 to 149°F)

Board dimensions: 99 mm x 219 mm

Weight: 210 gm (7.42 oz.)

CHAPTER 2

INSTALLATION

INSTALLATION

Initial Inspection

When you receive your PCL-813 you should find enclosed:

- One PCL-813 32-Channel Single-Ended Isolated Analog Input Card
- User's Manual
- Utility Diskette, which includes the card's software driver

The PCL-813 was carefully inspected both mechanically and electrically before shipment. It should be free of marks and scratches and in perfect electrical order on receipt.

When unpacking, check the unit for signs of shipping damage (damaged box, scratches, dents, etc). If there is damage to the unit or it fails to meet specifications, notify our service department or your local sales representative immediately. Also, call the carrier immediately and retain the shipping carton and packing material for the inspection by the carrier. We will make arrangements to repair or replace the unit.

Remove the PCL-813 card from its protective packaging carefully. Keep the anti-vibration package. Whenever you are not using the board, please store it in the package for protection.

Discharge any static electricity by touching the back of the system unit before you handle the board. You should avoid contact with materials that create static electricity such as plastic, vinyl, and styrofoam. The board should be handled only by the edges to avoid static electric discharge which could damage the integrated circuits on the PCL-813.

Warning! *Discharge your body's static electric charge by touching the back of the grounded chassis of the system unit (metal) before handling the board. You should avoid contact with materials that hold a static charge such as plastic, vinyl and styrofoam. Handle the board only by its edges to avoid static damage to its integrated circuits. Avoid touching the exposed circuit connectors.*

Switch and Jumper Settings

The PCL-813 has been designed with ease-of-use in mind. On-board the card you will notice that there is only one DIP switch (SW1). This switch is used to set the PCL-813's base address. Unipolar or Bipolar mode inputs is selected by jumper JP100. The following section goes into this in more detail.

I/O Address Selection

Most peripheral devices and interface cards are controlled by your PC's I/O ports. These devices and cards should be placed in an appropriate I/O space so that there will be no conflicts between them and the PCL-813. Keep in mind that the PCL-813 uses 16 consecutive address locations in your PC's I/O space. The following table provides an I/O port address map for your reference. This will assist in locating appropriate addresses for your other peripheral devices and interface cards. I/O port base addresses are selected from the 6-position DIP switch SW1 on-board the PCL-813. The valid addresses are 000 to 3F0 (hexadecimal). The factory default base address setting is 220. From time to time, you may find that you will have to use some of these spaces for other devices. If this is the case, then you can change the address according to the information given in the following table.

CARD I/O Address (SW1)

I/O ADDRESS RANGE	SWITCH POSITION					
	1	2	3	4	5	6
(HEXADECIMAL)	A9	A8	A7	A6	A5	A4
000 - 100	●	●	●	●	●	●
100 - 10F	●	○	●	●	●	●
------	-	-	-	-	-	-
------	-	-	-	-	-	-
200 - 20F	○	●	●	●	●	●
* 210 - 21F	○	●	●	●	●	○
220 - 22F	○	●	●	●	○	●
------	-	-	-	-	-	-
300 - 30F	○	○	●	●	●	●
------	-	-	-	-	-	-
3F0 - 3FF	○	○	○	○	○	○

○= Off ● = On *= Factory default setting

NOTE: A4 through A9 corresponds to your PC's address lines

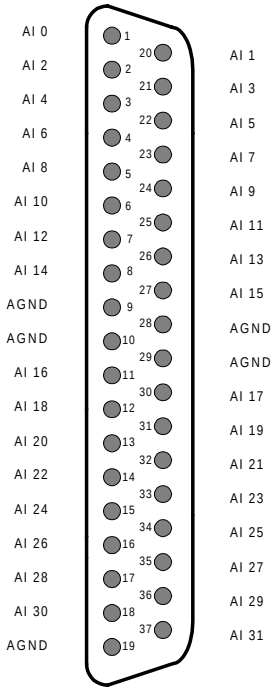
Input Range Selection

If your application always uses Unipolar input types, you should switch the JP100 to the "U" location, so the PCL-813 may accept to 0~10V, 0~5V, 0~2.5V and 0~1.25V Unipolar analog inputs.

Connector Pin Assignment

The PCL-813 is equipped with one 37-pin D-type connector located on the card's mounting bracket.

An illustration of this connector is given below:



Key: AI = Analog input

AGND = Analog ground

Plugging the PCL-813 into your PC

Warning! *Turn off your PC's power supply whenever you install or remove the PCL-813 or its cables. Static electricity can easily damage computer equipment. Ground yourself by touching the chassis of the computer (metal) before you touch any boards. See the static warning on page 6.*

Before you plug the PCL-813 into your PC, make sure that the computer's power is turned off, and that all power cords and all peripheral devices have been disconnected from the system.

Use the following procedure as a guideline for plugging the PCL-813 into your PC.

1. Remove the cover from the PC's chassis, and locate a vacant expansion slot on the passive backplane or motherboard for installing the PCL-813.
2. Take the card and insert it into the expansion slot, pressing it firmly into place. Use the card's mounting bracket as a guide between the chassis' rear panel and backplane or motherboard.
3. Once you have inserted the card firmly into the slot, secure it to the chassis by fastening its mounting bracket with a screw.
4. Attach the appropriate cable to connector CN1.
5. Replace the chassis cover, and reconnect all power cords and peripheral cables.

Installation of the PCL-813 is now complete.

CHAPTER 3

Signal Connection

SIGNAL CONNECTION

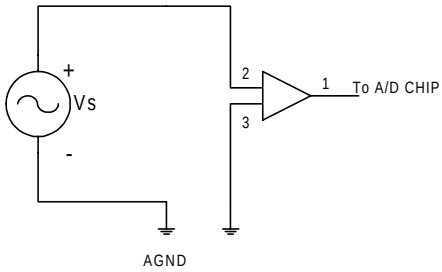
Since most data acquisition applications involve voltage measurement, it is important to make the correct signal connections in order to avoid any damage to your system, and to ensure accurate measurements. This chapter provides some helpful information about making the proper signal connections for your application.

Analog Input Connection

As you already know, the PCL-813 supports 32 single-ended isolated analog inputs. A single-ended analog input connection uses only one signal wire connected to an analog input terminal, which is referenced to a common ground. For example, in order to measure the voltage of a battery, simply connect its negative side to the PCL-813's ground (any one of the AGND pins on connector CN1), and its positive side to one of the card's analog input channels.

NOTE:
The PCL-813 does not support differential signal source inputs.

The following diagram illustrates a single-ended, common ground, analog input connection:



Single-Ended Analog Input Connection

CHAPTER 4

Register Structure And Format

REGISTER STRUCTURE AND FORMAT

This chapter has been written for those who wish to write their own software driver instead of using that of the PCL-813. The PCL-813 requires 16 consecutive addresses in I/O space. The most important issue in programming the PCL-813 is understanding the meaning of the 16 registers addressable from the selected I/O port base address. Here, you will find detailed information about the PCL-813's register formats and control procedures.

I/O Port Address Map

The following table shows you which base I/O addresses are used by the PCL-813. Refer to this map from time to time in order to become familiar with each of the card's register formats and their purpose. Sixteen consecutive registers corresponding to their I/O addresses are used to control the PCL-813's various functions. The following table has been provided in this chapter as a preface which outlines these addresses, relative to their location and control (read or write) assignments.

I/O Port Address Map

LOCATION	READ	WRITE
BASE+0	N/U	N/U
BASE+1	N/U	N/U
BASE+2	N/U	N/U
BASE+3	N/U	N/U
BASE+4	A/D low byte	N/U
BASE+5	A/D high byte	N/U
BASE+6	N/U	N/U
BASE+7	N/U	N/U
BASE+8	N/U	N/U
BASE+9	N/U	Gain control
BASE+10	N/U	Multiplexer scan control
BASE+11	N/U	N/U
BASE+12	N/U	Software A/D trigger
BASE+13	N/U	N/U
BASE+14	N/U	N/U
BASE+15	N/U	N/U

NOTE: N/U = Not used

The sections that follow provide further information about each register's data format according to its specific operation.

A/D Data Registers

The PCL-813 uses the data registers located at I/O ports BASE+4 and BASE+5 to store the converted A/D data. The low byte data is stored at BASE+4, and the high byte data is stored at BASE+5.

BASE+4 A/D Low Byte Data (Read)

D7	D6	D5	D4	D3	D2	D1	D0
AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

BASE+5 A/D High Byte Data (Read)

D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	DRDY	AD11	AD10	AD9	AD8

Where:

AD0 through AD11: Represent the PCL-813's A/D data bits. AD0 is the Least Significant Bit (LSB), and AD11 is the Most Significant Bit (MSB).

DRDY: Data Ready Bit. When A/D conversion is in progress, this bit remains as 1. It becomes 0 when the A/D conversion is completed.

Gain Control Register

BASE+9 is used to set the PCL-813's amplification gain for A/D conversion. The PCL-813 provides four different gains: x1, x2, x4, and x8.

The following tables outline BASE+9's register format and corresponding gain settings:

BASE+9 Gain Control Register (Write)

D7	D6	D5	D4	D3	D2	D1	D0
-	-	-	-	-	-	G1	G0

If JP100 is located at "B"

G1	G0	Gain	Input Range
0	0	x1	±5 V
0	1	x2	±2.5 V
1	0	x4	±1.25 V
1	1	x8	±0.625 V

If JP100 is locatesd at "U"

G1	G0	Gain	Input Range
0	0	x1	0-10V
0	1	x2	0-5V
1	0	x4	0-2.5V
1	1	x8	0-1.25V

Multiplexer Scan Register

The PCL-813 can multiplex up to 32 channels of analog input. Users have to set this register, located at BASE+10, to select the channel to be measured before performing any A/D conversion. This is done by selecting the channel to be used using a 5-bit register. The register format is as shown below:

BASE+10 Multiplexer Scan Control (Write)							
D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
-	-	-	C4	C3	C2	C1	C0

In order to facilitate the selection of 32 channels using the 5-bit register, each bit (D0-D4) can be set via the following table.

32 channel selection					
C4	**C3**	**C2**	**C1**	**C0**	**CH**
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	2
0	0	0	1	1	3
0	0	1	0	0	4
0	0	1	0	1	5
0	0	1	1	0	6
0	0	1	1	1	7
0	1	0	0	0	8
.	.	.	.	.	.
.	.	.	.	.	.
1	1	1	1	1	31

How to Initiate an A/D Conversion

The PCL-813 A/D conversion is software controlled, and is based upon the polling concept. After the A/D converter has been triggered, the application program checks the Data Ready Bit (DRDY) of the A/D status register. If this bit is detected (high), then A/D conversion is in progress. When this bit returns to low, then A/D conversion is completed, and the converted (binary) data may be read by the program.

There are two software methods to force the A/D converter to execute a single conversion: by writing a program yourself that writes the instruction directly to the I/O port, or by writing a program which utilizes that provided by the PCL-813 driver.

We suggest that you use the latter, the PCL-813 software driver, since:

- It makes your programming job easier.
- You will obtain a more readable source code which is easy to debug.
- It will enhance your program's performance.

For those who prefer to use the first method, writing directly to the I/O ports from within their own application, we provide the following step by step procedure:

Step 1: Set the desired channel by specifying the MUX scan range. Wait at least 5 ms before issuing a new command to the PCL-813.

Step 2: Trigger the A/D conversion by writing any value to BASE+12 of the A/D port, and then wait at least 20ms.

Step 3: Wait until the Data Ready Signal (Read the A/D high byte register DRDY bit, D4 of BASE+5) has returned to low.

Step 4: Obtain the binary data by reading the A/D registers (BASE+5 and BASE+4). First read the high byte, then the low byte.

Step 5: Convert the binary data to an integer value.

Refer to the DEMO01.C program for further details.

CHAPTER 5

Software

SOFTWARE

Introduction

This chapter describes the functions supported by the PCL-813 software driver. Programming languages supported by the driver include BASICA, GWBASIC, Quick BASIC 4.0/4.5, Microsoft C, Turbo C, Borland C++, Microsoft PASCAL, and Turbo PASCAL. The driver is a **Terminate and Stay Resident (TSR)** program which runs in the background while your application runs in the foreground. You must install the driver before you can use the following functions. Each PC-LabCard has its own driver, loaded by typing the appropriate filename at the DOS prompt. In the case of the PC-LabCard PCL-813, just type "PCL-813" at DOS prompt, and press enter to load the program.

Parameter Table

The software driver simplifies your programming by using a Parameter Table reference algorithm. The tables hold parameters, minimum and maximum values, and other specific information regarding the functions. In contrast, the application program contains tables specifying parameters and modes of operations. All the function calls supported by the drivers need only two arguments: the function number and a memory address pointer which points to a pre-defined Parameter Table.

Once the Parameter Table is defined, just assign the desired function number and the Parameter Table's address to the driver. Once this is done, it will pick up the necessary parameters associated with that specific function call, and then automatically execute the function.

A Parameter Table's format will be illustrated in detail later in this manual. What you have to know here is that the Parameter Table includes all the parameter settings necessary for all data acquisition function calls supported by the software driver. The following is an example of parameter tables:

Example (C language):

```
extern pcl813(int, unsigned int *);
unsigned param[60];          /* Parameter Table */
unsigned buffer[100];        /* A/D data buffer */

main()

{

unsigned far *ptr;           /* buffer pointer */

tr = (unsigned far *) buffer;
param[0] = 0;                /* card number      */
param[1] = 0x220;            /* I/O base address */
param[10] = FP_OFF(ptr);     /* offset address   */
param[11] = FP_SEG(ptr);     /* segment address  */
param[12] = 0;               /* only one buffer used
                             */
param[13] = 0;
param[14] = 100;             /* number of A/D conver-
                             sion  */
param[15] = 0x0;             /* A/D start channel*/
param[16] = 0xA;             /* A/D stop channel */
param[17] = 0x0;             /* gain code        */

pcl813(3,param);             /* initialize the PC-
                             LabCard */
pcl813(4,param);             /* initialize A/D func
                             tion   */
pcl813(5,param);             /* A/D conversions, and
                             store*/

                             /* converted data to
                             buffer  */
.
.
}
```

There are two ways to change your program's settings:

1. Modify corresponding parameters directly.

If you want to change the A/D start channel number, for example, you do not need to issue any function calls to change this setting, just change the corresponding parameter in your Parameter Table.

Example (C language):

```
extern pcl813(int, unsigned int *);
unsigned int param[60];/* Parameter Table*/

main()
{
.
.
.

param[15] = 0x0;      /* A/D start channel       */
param[16] = 0xA;      /* A/D stop  channel       */
pcl813(5,param);      /* S/W triggered A/D
                      conversion          */

.
.
param[15] = 0x2;      /* A/D start channel        */
pcl813(5,param);      /* S/W triggered A/D
                      conversion          */

.
.
}
```

2 Create a new Parameter Table.

The software driver's job-oriented algorithm gives your program the capability of addressing several Parameter Tables using the same function call or group of function calls in one program. It should be noted, however, that the driver can only address one Parameter Table at a time. The driver executes the jobs according to the specified Parameter Table. For convenient programming, we can define an individual Parameter Table in advance for each of frequently called functions without troublesome table modifications.

Example (C language):

```
extern pcl813(int, unsigned int *);
unsigned param1[60];/* Parameter Table 1  */
unsigned param2[60];/* Parameter Table 2  */

main()
{
.
.
.
/* JOB 1 */
param1[15] = 0x0; /* A/D start channel    */
param1[16] = 0xA; /* A/D stop  channel    */
param1[17] = 0x0; /* gain code            */
pcl813(5,param1); /* S/W triggered A/D conversion */
.
.
.
/* JOB 2 */
param2[15] = 0x2; /* A/D start channel    */
param2[16] = 0x8; /* A/D stop  channel    */
param2[17] = 0x5; /* gain code                  */
pcl813(5,param2); /* S/W triggered A/D conversion
*/
.
.
.
}
```

Two Parameter Tables are defined in this example, Jobs 1 and 2 are the same, except for the start channel, stop channel and gain setting.

NOTE:

1). When using BASIC language, negative numbers must be used to represent integer data over 32767.

2). Negative number = integer - 65536.

3). For example, you need to pass 45000 as a input parameter to BASIC function: 45000 - 65536 = -20536. So you have to use -20536 rather than 45000 as a parameter for the BASIC function.

Parameter List

Name	Size	Index
Board number	1 word	Param[0]
Base I/O address	1 word	Param[1]
Reserved	1 word	Param[2]
Reserved	1 word	Param[3]
Reserved	1 word	Param[4]
Reserved	2 words	Param[5]
Reserved	1 word	Param[7]
Reserved	1 word	Param[8]
Reserved	1 word	Param[9]
A/D Data Buffer A's address	2 words	Param[10]
A/D Data Buffer B's address	2 words	Param[12]
A/D conversion number	1 word	Param[14]
A/D start channel	1 word	Param[15]
A/D stop channel	1 word	Param[16]
Overall gain code	1 word	Param[17]
Gain code array pointer	2 words	Param[18]
Error number	1 word	Param[45]
Return value 0	1 word	Param[46]
Return value 1	1 word	Param[47]

Parameter Descriptions

param[0] 0 = Specify Number one Card.

1 = Specify Number two Card.

The software driver supports up to two PC-LabCards at one time. Set Param[0] to tell the driver which card is specified.

param[1] PC-LabCard I/O address can be anywhere from 200 (Hex) to 3F0 (Hex). The base address can be set to 200 or 210, 230 or 240 .... 3F0.

param[10] Offset address for A/D data buffer A.

param[11] Segment address for A/D data buffer A.

param[12] Offset address for A/D data buffer B.

param[13] Segment address for A/D data buffer B.

NOTE: For C or PASCAL, use their built-in memory allocation functions to allocate sufficient memory for buffers A and B. These memory allocation functions will return the offset and segment addresses. Save them to Param[10] through Param[13]. If buffer B is not used, be sure that you set Param[12] and Param[13] to 0.

Because BASICA and Quick BASIC do not provide memory allocation functions, you will have to assign explicit segment addresses for each buffer. If you assign a segment address as 0, then the driver will use the current data segment (DS) for buffers A and B. If buffer B is not used, be sure that you set Param[12] and Param[13] to 0.

param[14] This parameter sets the A/D conversion number. The range is from 1 to 32767.

param[15] Sets the A/D start channel number.

param[16] Sets the A/D stop channel number.

param[17] The driver allows you to set all the A/D channels to the same amplification gain. Param[17] sets the A/D gain code for all channels. Remember that individual amplification gains can be set for each channel, defined in the gain array table. This parameter is used only for setting all A/D channels to the same amplification gain. Set Param[17] to FF (Hex) to cause the driver to refer to the gain array table.

param[18] Offset address for the gain array table.

param[45] Error number.

param[46] Return value 0.

param[47] Return value 1.

Function List

Special Function Calls

Function 0: Get Error Message.

Function 1: Reserve.

Function 2: Get Driver Version Number.

Function 3: Driver Initialization.

A/D Function Calls

Function 4: A/D Initialization

Function 5: Perform A/D conversion with software data transfer.

Function Description

Special Function Calls

Function 0 : Get Error Message.

This function returns a zero-terminated text string pointer corresponding to an error code. The zero-terminated text string is a text string with numeric zero added at the end.

Parameters used:

param[45]: Error code.

param[46]: Offset of address of the string pointer.

param[47]: Segment of address of the string pointer.

Return data:

param[46]: Offset of address of the string pointer.

param[47]: Segment of address of the string pointer.

Function 2 : Get Driver Version Number.

This function returns the current version of the driver as well as the version of this Driver Specification.

Parameters used:

param[45]: Error code.

param[46]: Driver specification version number.

param[47]: Driver version number.

Return data:

param[45]: Error code.

param[46]: Driver specification version number.

param[47]: Driver version number.

Function 3: Driver Initialization.

The PC-LabCard is initialized according to the parameter's definitions. It will stop all functions and release all resources. It should be called before any other function.

Parameters used:

param[0]: Board number.

param[1]: Base I/O address. Return data.

param[45]: Error code.

A/D Function Calls

Function 4 : A/D Initialization

This function is used to initialize the PC-LabCard's A/D function according to the above parameter's setting.

Return data:

param[45]: Error code.

Function 5 : Perform A/D conversion with software data transfer.

This function will perform A/D conversion N times using software trigger with software data transfer. It will not return until the Nth. conversion has been completed. The value of 'N' is specified at param[14].

Return data:

param[45]: Error code.

Language Interface

BASICA

The following program example provides you with the appropriate procedures to load the language interface for BASICA and GWBASIC version 2.02.

Example:

```
100     CLEAR 49152!
110     DEF SEG = 0
120     SEG = 256 * PEEK(&H511) + PEEK(&H510)
130     SG = SG + 49152! / 16
140     DEF SEG = SG
150     BLOAD "813BAS.BIN", 0
```

GWBASIC (version 3.20)

The following program example provides you with the appropriate procedures to load the language interface for GWBASIC Version 3.20 and later.

Example:

```
110     'LOAD 813BAS.BIN DRIVER TO AN OUTSIDE AREA
120     DEF SEG= &H5000'DEFINE OUTSIDE AREA
130     BLOAD "813BAS.BIN"
140     'END OF DRIVER LOADING
```

QuickBASIC 4.0 and 4.5

The following program example provides you with the appropriate procedures to load the language interface for QuickBASIC 4.0 or 4.5.

Example 1:

```
QB filename /L 813QB.QLB
```

Example 2:

```
QB /L 813QB.QLB
```

Microsoft C

The following examples show you how to compile and link the interface for different modes using Microsoft C.

```
Small Mode:     Compile : cl /AS /c file.c
                Link    : link file+813CS.LIB;

Compact Mode:   Compile : cl /AC /c file.c
                Link    : link file+813CC.LIB;

Medium Mode:    Compile : cl /AM /c file.c
                Link    : link file+813CM.LIB;

Large Mode:     Compile : cl /AL /c file.c
                Link    : link file+813CL.LIB;
```

Turbo C

The following examples show you how to compile and link the interface for different modes using Turbo C.

DOS Command Line

Small Mode	:	TCC -ms file.c 813cs.lib
Compact Mode	:	TCC -mc file.c 813cc.lib
Medium Mode	:	TCC -mm file.c 813cm.lib
Large Mode	:	TCC -ml file.c 813cl.lib

Integrated Development Environment

You will need to use a general text editor to create a project file with the extension name "PRJ", for example 813.PRJ, which contains the corresponding mode interface and your program list.

Small Mode	:	The project file should contain 813cs.lib
Compact Mode	:	The project file should contain 813cc.lib
Medium Mode	:	The project file should contain 813cm.lib
Large Mode	:	The project file should contain 813cl.lib

Borland C++

The following examples show you how to compile and link the interface for different modes using Borland C++.

DOS Command Line

Small Mode	:	BCC -ms -c file.c 813cs.lib
Compact Mode	:	BCC -mc -c file.c 813cc.lib
Medium Mode	:	BCC -mm -c file.c 813cm.lib
Large Mode	:	BCC -ml -c file.c 813cl.lib

Integrated Development Environment

In Borland C++'s integrated environment, just pick the "Project" menu to create a project file and add the corresponding mode interface to it.

Microsoft PASCAL

The following examples show you how to compile and link the language interface using Microsoft PASCAL.

Example:

```
Compile  : pas1 demoxx pas2
Link     : link demoxx,,,813msp+pascal;
```

Turbo PASCAL

The following examples show you how to compile and link the language interface using Turbo PASCAL by adding certain statements to your program.

Example 1:

```
rogram main;

uses Crt;

{$F+}

{$L 813tpf} { use as far call }
```

Example 2:

```
rogram main;

uses Crt;

{$F-}

{$L 813tpn} { use as near call }
```

APPENDIX A

Calibration

CALIBRATION

In data acquisition and control applications, it is important to ensure that all measurement devices are calibrated regularly in order to maintain accuracy. A calibration program, CALB.EXE, is provided on the PCL-813 software disk to assist your calibration work.

The minimum equipment required to perform a satisfactory calibration is a 4½ digit digital multimeter. In addition, a voltage calibrator or stable DC voltage source is required. A card-extender, such as the PC-LabCard Model PCL-757, is an inexpensive device that you will find greatly improves access to the board during calibration and will probably be useful for other applications.

Calibration is easily performed using the CALB.EXE program. This program will lead you through the calibration and set-up procedure with a variety of prompts and graphic displays directing you to the appropriate adjustments. Material in this section is brief and is intended for use in conjunction with the calibration program.

VR Assignments

The PCL-813 has four on-board VRs, which will allow you to make accurate calibration adjustments for each of the card's A/D functions. The location of each VR is indicated in Figure A-1. The function of each VR is listed below:

VR1	A/D Bipolar offset adjustment
VR2	A/D Bipolar full-scale adjustment
VR3	Programmable amplifier offset adjustment
VR4	A/D Unipolar full-scale adjustment

PCL-813 32 CH S.E ISOLATED A/D CARD
JP 100
VR4
VR1 VR2 V
TP 1
U 19
U 21
SW1
(A987 654)

VR Location

A/D Calibration

Because the PCL-813 supports a variety of A/D input ranges, accurate calibration for a certain A/D range may result in a small offset when the input range is altered. It is strongly suggested that you recalibrate whenever a different A/D range is selected.

Calibration Steps:

a. Bipolar Adjustment: (JP100 located at "B")

(1) Short the A/D input of Channel 0 to AGND. Adjust VR1 until the reading of the A/D conversion flickers between 2047 and 2048.

(2) Apply a voltage with a full-scale value corresponding to the specific A/D input range to A/D Channel 0. Adjust VR2 until the reading of the A/D conversion flickers between 4094 and 4095.

b. Unipolar Adjustment: (JP100 located at "U")

(1) Short the A/D input of channel 0 to AGND. Adjust VRI until the reading of the A/D conversion flickers between 0000 and 0001.

(2) Apply a voltage with a full-scale value corresponding to the specific A/D input range to A/D channel 0. Adjust VR4 until the reading of the A/D conversion flickers between 4094 and 4095.

APPENDIX B

INDUSTRIAL TERMINATION BOARD

Introduction

The PCLD-881 is an universal screw terminal board designed for feild signal wiring in industrial applications. It can be connected to the analog and digital ports of various PC-LABCards via shielded cable and DB-37 connector.

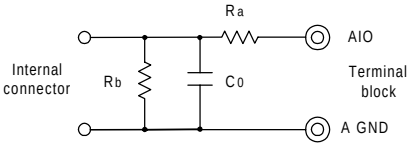
Due to the PCLD-881's special PCB layout you can install passive components to construct your own signal-conditioning circuits. You can easily construct a low-pass filter, attenuator or current-to-voltage converter by adding resistors and capacitors onto the board's circuit pads.

Features

- Low-cost universal screw-terminal board for PC-LabCards with 20-pin connectors
- 40 terminal points for one DB-37 port
- Reserved space for signal-conditioning circuits such as low-pass filter, current shunt and voltage attenuator
- Industrial type termination blocks permit heavy-duty and reliable connections of signals
- Table-top mounting using nylon standoffs. Screws and washers provided for panel or wall mounting
- Dimensions: 8.7" (L) x 4.53" (W) (221 mm x 115 mm)

Applications

- Field wiring for analog and digital I/O channels of PC-LabCards which employ standard or DB-37 connectors
- Signal-conditioning circuits can be implemented as illustrated in the following examples:



a) Straight-through connection (factory setting):
R_a = 0 W jumper
R_b = none (open)
C_0 = none(open)

b) 1.6 KHz (3 dB) low pass filter:

$$f_{3\,dB} = \frac{1}{2\pi R_a C_0}$$

R_a = 10 KW
R_b = none
C_o = 0.01 mF

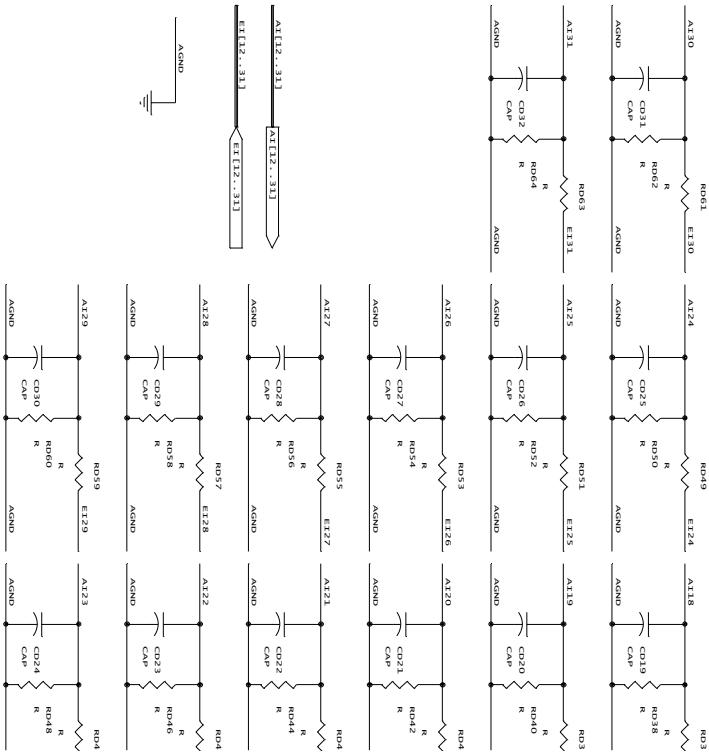
c) 10:1 voltage attenuator:

$$\text{Attenuation} = \frac{Rb}{Ra + Rb}$$

R_a = 9 KW
R_b = 1 KW
C_0 = none
(Assume source impedance $\ll$ 10 KW)

d) 4-20 mA to 1-5 V_{DC} signal converter:
Ra = 0 W jumper
R_b = 250 W (0.1% precision resistor)
C_0 = none

PCLD-881 Circuit Diagram



PCLD-881 Circuit Diagram

APPENDIX CALIBRATION

In data acquisition and control applications, it is important to ensure that all measurement devices are calibrated regularly in order to maintain accuracy. A calibration program, CALB.EXE, is provided on the PCL-813 software disk to assist your calibration work.

The minimum equipment required to perform a satisfactory calibration is a 4½ digit digital multimeter. In addition, a voltage calibrator or stable DC voltage source is required. A card-extender, such as the PC-LabCard Model PCL-757, is an inexpensive device that you will find greatly improves access to the board during calibration and will probably be useful for other applications.

Calibration is easily performed using the CALB.EXE program. This program will lead you through the calibration and set-up procedure with a variety of prompts and graphic displays directing you to the appropriate adjustments. Material in this section is brief and is intended for use in conjunction with the calibra-

tion program.

VR Assignments

The PCL-813 has four on-board VRs, which will allow you to make accurate calibration adjustments for each of the card's A/D functions. The location of each VR is indicated in Figure A-1. The function of each VR is listed below:

VR1	A/D Bipolar offset adjustment
VR2	A/D Bipolar full-scale adjustment
VR3	Programmable amplifier offset adjustment
VR4	A/D Unipolar full-scale adjustment

Figure A-1 VR Location

A/D Calibration

Because the PCL-813 supports a variety of A/D input ranges, accurate calibration for a certain A/D range may result in a small offset when the input range is altered. It is strongly suggested that you recalibrate whenever a different A/D range is selected.

Calibration Steps:

a. Bipolar Adjustment: (JP100 located at “B”)

(1) Short the A/D input of Channel 0 to AGND. Adjust VR1 until the reading of the A/D conversion flickers between 2047 and 2048.

(2) Apply a voltage with a full-scale value corresponding to the specific A/D input range to A/D Channel 0. Adjust VR2 until the reading of the A/D conversion flickers between 4094 and 4095.

b. Unipolar Adjustment: (JP100 located at “U”)

(1) Short the A/D input of channel 0 to AGND. Adjust VRI until the reading of the A/D conversion flickers between 0000 and 0001.

(2) Apply a voltage with a full-scale value corresponding to the specific A/D input range to A/D channel 0. Adjust VR4 until the reading of the A/D conversion flickers between 4094 and 4095.